~~Patent Literature Abstracts

8/3,K/1 (Item 1 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0017350069 *Drawing available*
WPI Acc no: 2008-B70508/200812
XRPX Acc No: N2008-135089

Computer code object runtime execution dynamically optimizing method, involves promoting future request for permission into permission assertion if permission is optimized, and continuing execution if permission is not optimized

Patent Assignee: NOVELL INC (NOVE-N)
Inventor: POULIOT S

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20080028461	A1	20080131	US 2006493010	A	20060726	200812	B

Priority Applications (no., kind, date): US 2006493010 A 20060726

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 20080028461	A1	EN	9	4	

Alerting Abstract ...NOVELTY - The method involves allowing requested **permission** for sequential stack frames in a call stack, and evaluating security elements associated with the requested **permission** and each stack frame of the call stack for determining whether the **permission** can be optimized. A future request is automatically promoted for the same **permission** into a **permission** assertion, if the **permission** is optimized. Execution is continued if the **permission** is not optimized.

Original Abstracts: The invention relates to a system and method for efficient security runtime. If the same security demand for **permissions** occurs twice during the same code path (i.e. execution stack) the latter can be automatically turned (optimized) into a security assertion based on the... ... stack frame. If the method being called has been allowed to execute before then a demand may be replaced with an assertion for the same **permissions** within the call stack. If that frame was executed then it means the security demand was successfully evaluated. Furthermore, if the **permission** evaluation result is known to be static (e.g., its result will not change) it can be determined that another check on the same **permissions** is not required higher on the stack, so this demand can safely be replaced by an assertion, which can effectively speed up the code execution...

...**Claims:**1. A computer-implemented method of dynamically optimizing runtime execution of computer code object on a computer by applying stack manipulation techniques, comprising:performing a **stack walk** for evaluating whether a requested **permission** is allowed for the plurality of sequential stack frames of a call stack;if the requested **permission** is not allowed for at least one of the sequential stack frames in the call stack, then creating a security exception;if the requested **permission** is allowed for the plurality of sequential stack frames in the call stack, then evaluating security elements associated with the requested **permission** and each stack frame of the call stack for determining whether the **permission** can be optimized;if yes, then automatically promoting a future request for the same **permission** into a **permission** assertion;if not, then continuing with the execution.

^ 8/3,K/2 (Item 2 from file: 350) (same assignee)
DIALOG(R)File 350: Derwent WPIX
(c) 2010 Thomson Reuters. All rights reserved.
0015995406 *Drawing available*
WPI Acc no: 2006-527076/200654
XRPX Acc No: N2006-422054

Called code frame execution determination method involves determining whether requested permission is associated with code assembly, responsive to demanding operation

Patent Assignee: MICROSOFT CORP (MICT)
Inventor: FEE G D; KAMATH A C; KOHNFELDER L M; LAMACCHIA B A

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 7076557	B1	20060711	US 2000613032	A	20000710	200654	B

Priority Applications (no., kind, date): US 2000613032 A 20000710

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 7076557	B1	EN	23	6	

Alerting Abstract ...NOVELTY - The method involves dynamically overriding a set of **permissions** assigned to a **permission** grant object associated with a code assembly preceding another code assembly. The requested **permission** is determined whether it is associated with the code assembly, responsive to the demanding operation. Execution of called code frame is permitted to perform protected operation, if the requested **permission** is provided in association with the code assembly. ... computer program product for determining whether requested **permission** is satisfied within runtime call stack; and runtime system for determining whether requested **permission** is satisfied within runtime call stack... ... USE - For determining whether requested **permission** for executing called code frame, is within runtime call stack...
Original Abstracts:A system and method determine whether a called code frame has a requested **permission** available to it, so as to be able to execute a protected operation. A code frame is contained within a code assembly received from a remote or local resource location. A **policy** manager generates a **permission** grant set containing **permission** grant objects associated with the code assembly. Both the **permission** grant set and the code assembly are loaded into a runtime call stack for runtime execution of one or more code frames. Calls to other code frames may involve loading additional code assemblies and **permission** grant sets into the runtime call stack. In order for a called code frame to perform a protected operation, the code frame demands a requested **permission** from its calling code frame and all code frames preceding the calling code frame on the runtime call stack as part of a **stack walk** operation. If the calling code frame and the preceding call frames can satisfy the requested **permission**, the called code frame can perform the protected operation (absent stack overrides). Otherwise, a security exception is thrown and the called code frame is inhibited from performing the protected operation (absent stack overrides). Stack overrides may be employed to dynamically modify the **stack walk** operation. To increase performance, a **stack walk** may be avoided by caching an intersection of the **permission** grants of all code assemblies in the application. **Claims:**We

claim:1. A method of determining whether a requested **permission**, wherein the **permission** is at least one of a set of **permissions**, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the method comprising:associating a first **permission** grant object with a first code assembly in the runtime call stack;dynamically overriding the set of **permissions** that is assigned to a second **permission** grant object associated with a second code assembly preceding the first code assembly;creating a **permission** request object within the called code frame to demand the requested **permission**;demanding via the **permission** request object the requested **permission** from the first **permission** grant object to allow the called code frame to perform the protected operation;determining whether the requested **permission** is provided in association with the first code assembly by the first **permission** grant object, responsive to the demanding operation; andpermitting the execution of the called code frame to perform the protected operation, if the requested **permission** is provided in association with the first code assembly, whereby a full walk of the runtime call stack may be avoided.

11/3,K/2 (Item 2 from file: 350)
DIALOG(R)File 350: Derwent WPIX

0018768268 *Drawing available*
WPI Acc no: 2009-F48516/200917
Computer-implemented program i.e. software program, optimization performing method for post-link monitoring and optimization tool, involves optimizing modified program code responsively to clone-specific profile data
Patent Assignee: HABER G (HABE-I); LEVIN R (LEVI-I); UR S (URSS-I)
Inventor: HABER G; LEVIN R; UR S

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20090055813	A1	20090226	US 2007842180	A	20070821	200917	B

Priority Applications (no., kind, date): US 2007842180 A 20070821

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 20090055813	A1	EN	16	7	

Alerting Abstract ...a function in a program code, and cloning the function to create a modified program code having a set of instances of the function. Call **paths** of the function are distributed in a modified program code to assign respective modified call **path** to each instance of the function. The modified program code is executed while accumulating respective clone-specific profile data for the instances of the function... ... of each thread in order to determine the calling function of the executed instruction. The method utilizes efficient profiling technique, and avoids the overhead of **stack walking** at each sampled event or instruction. The method enables collection of calling context hardware events easily...

...**Original Abstracts:**profiling methods such as hardware event sampling, basic block profiling, and edge profiling may then be applied to the modified program code to obtain call

path-based, clone-specific profile data. The profile data can be further exploited to optimize the program code.

...**Claims:**of program optimization, comprising the steps of: identifying a function in program code, said function having call sites, said call sites each having respective call **paths** leading thereto; cloning said function to create a modified program code having a plurality of instances of said function therein; distributing said call **paths** of said function in said modified program code to assign a respective modified call **path** to each of said instances of said function; executing said modified program code while accumulating respective clone-specific profile data for said instances of said...

11/3,K/3 (Item 3 from file: 350)
DIALOG(R)File 350: Derwent WPIX
(c) 2010 Thomson Reuters. All rights reserved.
0016193356 *Drawing available*
WPI Acc no: 2006-724997/200675
XRPX Acc No: N2006-569838
Hybrid stack walking method of call stack, involves performing managed stack walk on call stack and native stack walk on native frames of call stack
Patent Assignee: MICROSOFT CORP (MICT)
Inventor: GOLDIN M; WIJERATNA T

Patent Family (2 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20060212844	A1	20060921	US 200583843	A	20050318	200675	B
US 7574702	B2	20090811	US 200583843	A	20050318	200953	E

Priority Applications (no., kind, date): US 200583843 A 20050318

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 20060212844	A1	EN	16	7	

Alerting Abstract ...NOVELTY - A managed **stack walk** is performed on a call stack comprising several managed frames and native frames associated with a mixed code. A native **stack walk** is performed on the native frames of the call stack, to obtain hybrid **stack walking**. ... computer readable medium comprising instructions for hybrid **stack walking**; and computer... ... USE - For identifying critical **paths** of call stack... ... ADVANTAGE - The hybrid **stack walking** is performed for assembling information about executing modules or functions in the code...

Original Abstracts: In one embodiment, a method and apparatus for **stack walking** a **call stack** associated with mixed code, by interleaving a native **stack walking** process with a managed **stack walking** process. Mixed code comprises at least one managed instruction and at least one native instruction, and the call stack comprises at least one managed frame... ... managed frames being associated with the managed instructions, and the native frames being associated with native instructions. The method comprises acts of performing a managed **stack walk** on the **call** stack, a native **stack walk** on native frames of the call stack. In a further embodiment, handling indirect jumps during a native **stack walk**, and in another embodiment, detecting validity of a memory address... ... In one embodiment, a method and apparatus for **stack walking** a **call stack** associated with mixed code, by interleaving a native **stack walking** process with a managed **stack walking** process. Mixed code comprises at least one managed instruction and at least one native instruction, and the call stack comprises at least one managed frame... ... managed frames being associated with the managed instructions, and the native frames being associated with native instructions. The method comprises acts of performing a managed **stack walk** on the **call** stack, a native **stack walk** on native frames of the call stack. In a further embodiment, handling indirect jumps during a native **stack walk**, and in another embodiment, detecting validity of a memory address.

Claims: What is claimed:**1**. A method of **stack walking** a **call stack** associated with mixed code, wherein the mixed code comprises at least one managed instruction and at least one native instruction, the call stack comprises at... ... the at least one native frame being associated with the at least one native instruction, the method comprising the acts of:(A) performing a managed **stack walk** on the **call** stack; and(B) performing a native **stack walk** on the at least one native frame of the call stack... ... What is claimed: 1. A method of **stack walking** a **call stack** associated with mixed code, wherein the mixed code comprises at least one managed instruction and at least one native instruction, the method comprising the acts... ... one native frame being associated with a second function having at least one native instruction, the native frame containing a second return address; performing a **stack walk** of the **call** stack to detect a managed frame on the call stack; in response to detecting a managed frame on the call stack, performing a managed **stack walk** of managed frames on the call stack to resolve the managed frames on the call stack; and after performing the managed **stack walk**, performing a native **stack walk** of native frames on the call stack to resolve the native frames on the call stack, including native frames between managed frames.

11/3,K/4 (Item 4 from file: 350)
DIALOG(R)File 350: Derwent WPIX
(c) 2010 Thomson Reuters. All rights reserved.
0014202755 *Drawing available*
WPI Acc no: 2004-388356/200436
XRPX Acc No: N2004-309195

Call chain identification method in interrupted program, involves updating instruction and stack pointers based on distance variables on which selected calculations are performed

Patent Assignee: MICROSOFT CORP (MICT)
Inventor: PIERCE K B

Patent Family (2 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20040083460	A1	20040429	US 2002279550	A	20021023	200436	B
US 7178132	B2	20070213	US 2002279550	A	20021023	200714	E

Priority Applications (no., kind, date): US 2002279550 A 20021023

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 20040083460	A1	EN	30	26	

Alerting Abstract ...stack with return address, stack and instruction pointers is received. The calculations to be performed on distance variables are selected based on instructions identified on **path** of instructions. The selected calculations are performed on variables. The instruction and stack pointers are updated using calculated variables, and list of instruction pointer is ...

...**Claims:**while the call stack still contains return addresses, performing the following,following the control flow in a binary image, from the instruction pointer, through a **path** of instructions, to a return instruction;selecting calculations to perform on distance variables based on instructions identified in the **path** of instructions;performing the selected calculations on the distance variables;using the calculated distance variables to update the instruction pointer and stack pointer; andreturning... ... comprising, a binary image with an associated stack frame;an interrupt program that interrupts the application program and saves the execution state, and calls a **stack walking** program; andthe **stack walking** program comprising, instructions for walking forward through binary images to identify instructions used to calculate offsets into the stack frame associated with the binary image...

11/3,K/5 (Item 5 from file: 350)
DIALOG(R)File 350: Derwent WPIX
(c) 2010 Thomson Reuters. All rights reserved.
0010093072 *Drawing available*
WPI Acc no: 2000-399802/200034
XRPX Acc No: N2000-299508
Computer system for Java language applications, has compiler which compiles fragment of code of particular application
Patent Assignee: ESMERTEC AG (ESME-N); INSIGNIA SOLUTIONS LTD (INSI-N); INSIGNIA SOLUTIONS PLC (INSI-N)
Inventor: ALEC DIAS B A; CHARNELL W; CHARNELL W T; DARNELL S; DIAS B; DIAS B A A; GUTHRIE P; GUTHRIE P J; KRAMSKOY J; KRAMSKOY J P; PLUMMER W; RAUTENBACH K ; RAUTENBACK K; SEXTON J; SEXTON J J; THOMAS S; THOMAS S P; WYNN M; WYNN M J; DIAS B A

Patent Family (28 patents, 84 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 2000029937	A2	20000525	WO 1999GB788	A	19990316	200034	B
AU 199928469	A	20000605	AU 199928469	A	19990316	200042	E
US 20020029357	A1	20020307	WO 1999GB788	A	19990316	200221	E
			US 2001859161	A	20010516		
US 20020032719	A1	20020314	WO 1999GB788	A	19990316	200222	E
			US 2001859163	A	20010516		
US 20020032822	A1	20020314	WO 1999GB788	A	19990316	200222	E
			US 2001859134	A	20010516		
US 20020040470	A1	20020404	WO 1999GB788	A	19990316	200227	E
			US 2001859162	A	20010516		
US 20020042807	A1	20020411	WO 1999GB788	A	19990316	200227	E
			US 2001858826	A	20010516		
US 20020049865	A1	20020425	WO 1999GB788	A	19990316	200233	E
			US 2001859135	A	20010516		

EP 1208425	A2	20020529	EP 1999909100	A	19990316	200243	E
			WO 1999GB788	A	19990316		
US 20020104077	A1	20020801	WO 1999GB788	A	19990316	200253	E
			US 2001858578	A	20010516		
US 20020108106	A1	20020808	WO 1999GB788	A	19990316	200254	E
			US 2001859072	A	20010516		
US 20020108107	A1	20020808	WO 1999GB788	A	19990316	200254	E
			US 2001859133	A	20010516		
US 20020112227	A1	20020815	WO 1999GB788	A	19990316	200256	E
			US 2001858827	A	20010516		
US 20020165848	A1	20021107	WO 1999GB788	A	19990316	200275	E
			US 2001858426	A	20010516		
JP 2003526135	W	20030902	WO 1999GB788	A	19990316	200358	E
			JP 2000582880	A	19990316		
US 6691303	B2	20040210	WO 1999GB788	A	19990316	200413	E
			US 2001859162	A	20010516		
US 6766513	B2	20040720	WO 1999GB788	A	19990316	200448	E
			US 2001859161	A	20010516		
US 6862728	B2	20050301	WO 1999GB788	A	19990316	200516	E
			US 2001859133	A	20010516		
US 6901587	B2	20050531	WO 1999GB788	A	19990316	200536	E
			US 2001859072	A	20010516		
US 6925637	B2	20050802	WO 1999GB788	A	19990316	200550	E
			US 2001858826	A	20010516		
US 7007005	B2	20060228	WO 1999GB788	A	19990316	200616	E
			US 2001858426	A	20010516		
US 7039738	B2	20060502	WO 1999GB788	A	19990316	200629	E
			US 2001859134	A	20010516		
US 7058929	B2	20060606	WO 1999GB788	A	19990316	200638	E
			US 2001859135	A	20010516		
US 7069549	B2	20060627	WO 1999GB788	A	19990316	200643	E
			US 2001858578	A	20010516		
US 7080366	B2	20060718	WO 1999GB788	A	19990316	200648	E
			US 2001858827	A	20010516		
US 20080016507	A1	20080117	WO 1999GB788	A	19990516	200807	E
			US 2001859163	A	20010516		
			US 2007771629	A	20070629		
EP 1208425	B1	20080903	EP 1999909100	A	19990316	200859	E
			WO 1999GB788	A	19990316		

DE 69939495	E	20081016	DE 69939495	A	19990316	200868	E
			EP 1999909100	A	19990316		
			WO 1999GB788	A	19990316		

Priority Applications (no., kind, date): GB 199825102 A 19981116

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 2000029937	A2	EN	203	12		
National Designated States,Original	AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZA ZW					
Regional Designated States,Original	AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SL SZ UG ZW					
AU 199928469	A	EN			Based on OPI patent	WO 2000029937
US 20020029357	A1	EN			Continuation of application	WO 1999GB788
US 20020032719	A1	EN			Continuation of application	WO 1999GB788
US 20020032822	A1	EN			Continuation of application	WO 1999GB788
US 20020040470	A1	EN			Continuation of application	WO 1999GB788
US 20020042807	A1	EN			Continuation of application	WO 1999GB788
US 20020049865	A1	EN			Continuation of application	WO 1999GB788
EP 1208425	A2	EN			PCT Application	WO 1999GB788
					Based on OPI patent	WO 2000029937
Regional Designated States,Original	DE ES FR GB IT SE					
US 20020104077	A1	EN			Continuation of application	WO 1999GB788
US 20020108106	A1	EN			Continuation of application	WO 1999GB788
US 20020108107	A1	EN			Continuation of application	WO 1999GB788
US 20020112227	A1	EN			Continuation of application	WO 1999GB788
US 20020165848	A1	EN			Continuation of application	WO 1999GB788
JP 2003526135	W	JA	308		PCT Application	WO 1999GB788
					Based on OPI patent	WO 2000029937
US 6691303	B2	EN			Continuation of application	WO 1999GB788
US 6766513	B2	EN			Continuation of application	WO 1999GB788
US 6862728	B2	EN			Continuation of application	WO 1999GB788
US 6901587	B2	EN			Continuation of application	WO 1999GB788
US 6925637	B2	EN			Continuation of application	WO 1999GB788
US 7007005	B2	EN			Continuation of application	WO 1999GB788
US 7039738	B2	EN			Continuation of application	WO 1999GB788

US 7058929	B2	EN			Continuation of application	WO 1999GB788
US 7069549	B2	EN			Continuation of application	WO 1999GB788
US 7080366	B2	EN			Continuation of application	WO 1999GB788
US 20080016507	A1	EN			Continuation of application	WO 1999GB788
					Continuation of application	US 2001859163
EP 1208425	B1	EN			PCT Application	WO 1999GB788
					Based on OPI patent	WO 2000029937
Regional Designated States,Original	DE ES FR GB IT SE					
DE 69939495	E	DE			Application	EP 1999909100
					PCT Application	WO 1999GB788
					Based on OPI patent	EP 1208425
					Based on OPI patent	WO 2000029937

Alerting Abstract ...NOVELTY - A compiler is configured to compile a fragment of the code of an application (24). The fragment of code is a dominant **path** fragment which comprises one or more blocks of code.
Original Abstracts:A computer system described may have features relating to one or more of dynamic compilation of a dominant **path**, including using pre-exception condition checks, outliers and/or class loaders, to dispatch mechanisms for interface methods, to management and deletion of code buffers, to... ... A method and system of memory management using **stack walking**. The method of managing memory in a computer system includes identifying compiled code to be deleted, examining the return addresses of the frames in the... dominant code blocks are stored in one portion of the memory and the outliers are stored in another portion of the memory. Storing the dominant **path** code separate from the outliers increases efficiency of the system... ... A dynamic compiler and method of compiling code to generate a dominate **path** and handle exceptions. The dynamic compiler includes an execution history recorder that is configured to record the number of times a fragment of code is interpreted... ... came from and where transfer of control goes to for each fragment of code that is executed, thereby allowing for compilation of a dominant **path** of code. If the execution of code deviates from the dominant **path** of compiled code (such as when an exception occurs), a fallback interpreter is utilized to interpret the fragment of code to be executed... A method and a system of memory management using **stack walking**. The method of managing memory in a computer system includes identifying compiled code to be deleted, examining the return addresses of the frames in the stack... ... code blocks are stored in one portion of the memory and the outliers are stored in another portion of the memory. Storing the dominant **path** code separate from the outliers increases efficiency of the system... A dynamic compiler and method of compiling code to generate a dominate **path** and handle exceptions. The dynamic compiler includes an execution history recorder that is configured to record the number of times a fragment of code is interpreted... ... came from and where transfer of control goes to for each fragment of code that is executed, thereby allowing for compilation of a dominant **path** of code. If the execution of code deviates from the dominant **path** of compiled code (such as when an exception occurs), a fallback interpreter is utilized to interpret the fragment of code to be executed... ... A computer system described may have features relating to one or more of dynamic compilation of a dominant **path**, including using pre-exception condition checks, outliers and/or class loaders, to dispatch mechanisms for interface methods, to management and deletion of code buffers, to test...

...**Claims:**a program during execution of a program, the method comprising the steps of: (a) first determining whether a first piece of code includes a dominant **path** therethrough formed of a series of program instructions for execution one after another in sequence during execution of the dominant **path**;(b) first determining whether the first piece of code includes a control transfer instruction therein;(c) first compiling the first piece of code by the... ... execution of the program to provide a first piece of compiled code only if the first piece of code is determined to be a dominant **path** and the control transfer instruction is determined to be present in the first piece of code;(d) second compiling a second piece of code by... ... a compiler manager coupled thereto and a threshold number of executions for a threshold comparison with a recorded number of times to determine a dominant **path** having a fragment to be compiled therein;a compiler queue of fragments to be compiled for receiving the fragment to be compiled and a successor... ... a received fragment to be compiled and the corresponding successor fragment of the received fragment to be compiled, and (ii) to create a compiled dominant **path** from the received fragment to be compiled, the corresponding successor fragment being compiled in accordance with its correspondence with the fragment to be compiled and... ... reached the threshold number of executions;the execution history recorder being further configured to record from where a transfer of control into the compiled dominant **path** came and to where control is transferred out of the compiled dominant **path**;a queue duration determination including (i) a determination whether the length of the compiler queue has exceeded a predetermined length and (ii) a determination whether...

~~Patent Literature Full-Text

DIALOG(R)File 348: EUROPEAN PATENTS

8/3K/1 (Item 1 from file: 348)
01383064

Unified data type system and method
Vereinheitlichtes Datentypsystem und Verfahren
Systeme et methode de type de donnees unifie

Patent Assignee:

- **MICROSOFT CORPORATION** (749861)
  One Microsoft Way; Redmond, Washington 98052-6399 (US)
  (Applicant designated States: all)

Inventor:

- **Bossworth, George H.**
  19830 NE 123rd Court; Woodinville,Washington 98072; (US)
- **Dussud, Patrick H.**
  6008 142nd Court SE; Bellevue,Washington 98006; (US)
- **Miller, James S.**
  17213 NE 4th Place; Bellevue,Washington 98008; (US)
- **Olander, Daryl B.**
  720 Juniper Ave.; Boulder,Colorado 80304; (US)

Legal Representative:

- **Grunecker, Kinkeldey, Stockmair & Schwanhausser Anwaltssozietat (100721)**
  Maximilianstrasse 58; 80538 Munchen; (DE)

	Country	**Number**	**Kind**	**Date**	
Patent	EP	1174791	A2	20020123	(Basic)
	EP	1174791	A3	20071219	
Application	EP	2001116860		20010710	
Priorities	US	613289		20000710	
	US	614158		20000711	

Specification: ...loads the files for execution. The loader 530 receives the executable file and resolves necessary references and loads the code. The environment may provide a **stack walker** 532, i.e., the piece of code that manages the method calls and provides for the identification of the sequence of method calls on a... ...to be executed. The execution environment may further provide a security module 536 to prevent unauthorized use of resources by determining whether certain code has **permission** to access certain system resources (or even execute at all). The runtime environment may further provide memory management services, such as a garbage collector 538...

DIALOG(R)File 348: EUROPEAN PATENTS

8/3K/2 (Item 2 from file: 348)
01233627

Method and architecture to support multiple services in label switched networks
Verfahren und Architektur zur Unterstuzung von mehreren Diensten in einem Etikettvermittlungsnetzwerk
Procede et architecture permettant des service multiples dans un reseau a commutation d' etiquette

Patent Assignee:

- **Nortel Networks Limited** (3029042)
  2351 Boulevard Alfred-Nobel; St Laurent, Quebec H4S 2A9 (CA)
  (Proprietor designated states: all)

Inventor:

- **Mauger, Roy Harold**
  47 Beech Avenue; Radlett,Hertfordshire WD7 7DD; (GB)
- **Brueckheimer, Simon Daniel**
  74 Church Street; London N10 3NE; (GB)

Legal Representative:

- **Hermele, Daniel Stephen et al (159941)**
  Nortel Networks Intellectual Property Law Group London Road; Harlow, Essex CM17 9NA; (GB)

	Country	**Number**	**Kind**	**Date**	
Patent	EP	1069742	A2	20010117	(Basic)
	EP	1069742	A3	20030924	
	EP	1069742	B1	20071212	
Application	EP	2000305973		20000713	
Priorities	US	354651		19990716	

Specification: ...similar messaging protocols. This modified version of SIP will be referred to in the following description as SIP++. An extension to the IETF Common Open **Policy** Service (COPS) in provides communication between the physical MPLS network and its control services. Call **Walkthrough** for Successful **Call**

A call walkthrough for a successful call is illustrated diagrammatically in figure 3. This figure corresponds to the level of detail shown in figure 1...

Specification: ...similar messaging protocols. This modified version of SIP will be referred to in the following description as SIP++. An extension to the IETF Common Open **Policy**

Service (COPS) in provides communication between the physical MPLS network and its control services. Call **Walkthrough** for Successful **Call**

A call walkthrough for a successful call is illustrated diagrammatically in figure 3. This figure corresponds to the level of detail shown in figure 1...

DIALOG(R)File 348: EUROPEAN PATENTS
(c) 2010 European Patent Office. All rights reserved.
14/3K/3 (Item 3 from file: 348)
00747354

Capability engine method and apparatus for a microkernel data processing system
Verfahren und Gerat mit Fahigkeitsvorrichtung fur ein Mikrokern-Datenverarbeitungssystem
Methode et appareil a dispositif de capacite pour un systeme de traitement de donnees a micro-noyaux

Patent Assignee:

- **International Business Machines Corporation** (200120)
  Old Orchard Road; Armonk, N.Y. 10504 (US)
  (Proprietor designated states: all)

Inventor:

- **Magee, James Michael**
  5310 Canal Drive; Lake Worth, Florida 33463; (US)
- **Sotomayor, Guy Gil, Jr.**
  6042 Sherwood Glen Way, Apt. 3; West Palm Beach, Florida 33415; (US)
- **Youngworth, Christopher Dean**
  3 Gulfview Court; Savoy, Illinois 61874; (US)

Legal Representative:

- **Williams, Julian David (75461)**
  IBM United Kingdom Limited, Intellectual Property Department, Hursley Park; Winchester, Hampshire SO21 2JN; (GB)

	Country	**Number**	**Kind**	**Date**	
Patent	EP	704796	A2	19960403	(Basic)
	EP	704796	A3	19980701	
	EP	704796	B1	20000419	
Application	EP	95304188		19950616	
Priorities	US	263313		19940928	

Specification: ...different address space.

The queuing mechanism and scheduling policies are associated with the port object and are not specific to the capability engine 300. The **specific** scheduling queuing **policy** the capability engine 300 will call may be altered on a port by port basis via calls to the capability engine 300. There are two...for the CONTROLLED placement of data into UNMAPPED portions of the task's address space, only MAPPED ones. UNMAPPED placement is supported through the simple **model** via a capability **call** on the target capability. There is currently no plan to include this option in the by-reference case as it can be mimicked by first... ...optimizations based on roll in of additional function is required, a separate new library should be created. This library is free to borrow interface and **execution path** notions from the message passing library 220, but is not obligated to do so. Such a library would operate on top of the capability engine...

DIALOG(R)File 348: EUROPEAN PATENTS

14/3K/4 (Item 4 from file: 348)
00738317

System and method for interprocess communication
System und Verfahren zur Kommunikation zwischen Prozessen
Systeme et methode pour la communication entre des processus

Patent Assignee:

- **International Business Machines Corporation** (200120)
  Old Orchard Road; Armonk, N.Y. 10504 (US)
  (applicant designated states: AT;BE;CH;DE;ES;FR;GB;IT;LI;NL;SE)

Inventor:

- **Faruqi, Aziza Bushra**
  1170 NW 15th Avenue No. 105; Boca Raton, Florida 33486; (US)
- **Green, Joseph William**
  301 Norwood Terrace No. 129; Boca Raton, Florida 33431-6588; (US)
- **Youngworth, Christopher Dean**
  No. 3 Gulfview Courts; Savoy, Illinois 61874-0363; (US)

Legal Representative:

- **Williams, Julian David (75461)**
  IBM United Kingdom Limited, Intellectual Property Department, Hursley Park;
  Winchester, Hampshire SO21 2JN; (GB)

Specification: ...different address space.

The queuing mechanism and scheduling policies are associated with the port object and are not specific to the capability engine 300. The **specific** scheduling queuing **policy** the capability engine 300 will call may be altered on a port by port basis via calls to the capability engine 300. There are two...allow for the CONTROLLED placement of data into UNMAPPED portions of the tasks address space, only MAPPED ones. UNMAPPED placement is supported through the simple **model** via a capability **call** on the target capability. There is currently no plan ... optimizations based on roll in of additional function is required, a separate new library should be created. This library is free to borrow interface and **execution path** notions from the message passing library 220, but is not obligated to do so. Such a library would operate on top of the capability engine...

~~Non-Patent Literature Abstracts

18/3,K/2 (Item 2 from file: 2)
DIALOG(R)File 2: INSPEC

10496331
Title: Information modeling for automated risk analysis
Author(s): Chivers, H.
Author Affiliation: Dept. of Inf. Syst., Cranfield Univ., Bedford, UK
Book Title: Communications and Multimedia Security. 10th IFIP TC-6 TC-11 International Conference, CMS 2006. Proceedings (Lecture Notes in Computer Science Vol. 4237)
Inclusive Page Numbers: 228-39
Publisher: Springer-Verlag, Berlin
Country of Publication: Germany
Publication Date: 2006
Conference Title: Communications and Multimedia Security. 10th IFIP TC-6 TC-11 International Conference, CMS 2006. Proceedings
Conference Date: 19-21 Oct. 2006
Conference Location: Heraklion, Crete, Greece
Editor(s): Leitold, H.; Markatos, E.
ISBN: 3 540 47820 5
Number of Pages: xii+251
Language: English
Subfile(s): C (Computing & Control Engineering); D (Information Technology for Business)
INSPEC Update Issue: 2007-025
Copyright: 2007, The Institution of Engineering and Technology
Abstract: ...balance complexity, scalability and expressiveness. This paper describes such a model; novel features include combining formal information modeling with informal requirements traceability to support the **specification** of **security requirements** on incompletely **specified** services, and the typing of information **flow** to quantify **path** exploitability and **model** communications security

18/3,K/3 (Item 3 from file: 2)
DIALOG(R)File 2: INSPEC

08073780
Title: MPLS management using policies
Author(s): Brunner, M.; Quittek, J.
Author Affiliation: C&C Res. Labs., NEC Europe Ltd, Heidelberg, Germany
Book Title: 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No.01EX470)
Inclusive Page Numbers: 515-28
Publisher: IEEE, Piscataway, NJ
Country of Publication: USA
Publication Date: 2001
Conference Title: Proceedings of 2001 International Symposium on Integrated Network Management
Conference Date: 14-18 May 2001
Conference Location: Seattle, WA, USA

Editor(s): Pavlou, G.; Anerousis, N.; Liotta, A.
ISBN: 0 7803 6719 7
U.S. Copyright Clearance Center Code: 0 7803 6719 7/2001/$10.00
Item Identifier (DOI): 10.1109/INM.2001.918063
Number of Pages: xxiv+886
Language: English
Subfile(s): B (Electrical & Electronic Engineering); C (Computing & Control Engineering)
INSPEC Update Issue: 2001-042
Copyright: 2001, IEE

Abstract: ...MPLS networks, which is crucial for large networks. We decided to follow the IETF Policy Framework approach and extended the common information model (CIM) for **policies** with MPLS **specific** classes. MPLS introduces the notion of a label switched path (LSP), possibly covering an entire network, which **calls** for an extension of the IETF Policy Framework into the direction of network and service management issues. We address this by preparing a three-level...

Identifiers: MPLS management; policies; multi-protocol label switching; standardization; Internet Engineering Task Force; traffic engineering; QoS; IP-networks; IETF Policy Framework approach; common information **model**; CIM; label switched **path**; LSP; service management; three-level policy architecture; network-level policies; policy-based management system

~~Non-Patent Literature Full-Text